

Web Development Lab 2 - Data Dashboards in Streamlit

CS 1301 - Intro to Computing - Fall 2025

Important

- Due Date: **Thursday, October 23rd, 11:59 PM.**
Early Submission (extra credit) **Monday, October 13th, 11:59 PM.**
- This is an individual assignment. High-level collaboration is encouraged, **but your submission must be uniquely yours.**
- Resources:
 - Installing Streamlit Handout
 - TA Helpdesk
 - Ed Discussion
- **This assignment is meant to be a self-guided exploration of web development. You are encouraged to search resources independently before seeking TA help.**

Purpose

In this lab, you will build another web app using the `streamlit` Python library, which can easily transform Python code into interactive web applications, dashboards, or reports that can be hosted online. There are three parts to this project that will put both your creative and analytical skills to the test. Enjoy!

Submission Overview

When you finish this assignment, you will need to zip all of your files into a compressed folder and submit to Gradescope under the correct assignment before the deadline (See [Submission Process](#) for more information). **Please double check your submission by re-downloading your submission from Gradescope to ensure that it is correct.** Please un-zip your downloaded file (see [Starter Files](#)) before you double check, since your webpage may not properly load inside of a compressed folder.

CS 1301 is not responsible for incorrect or missing submissions to Gradescope past the deadline of this lab. An incorrect or missing submission will receive no credit and is not eligible for a regrade request. PLEASE double check!

Document Overview

- Explanation
- Lab Rules
- Getting Started
 - Starter Files
 - Installing Streamlit
 - Additional Resources
- Part 1: Writing to a CSV File
- Part 2: Making a JSON
- Part 3: Reading Data and Making Graphs
- Extra Credit
- Submission Process
- Grading Rubric

Explanation

This lab is composed of three sections:

- In **Part 1**, you will create and write to a CSV file using data you collect from user input through your Streamlit Application.
- In **Part 2**, you will write data to a provided JSON file.
- In **Part 3**, you will use the CSV file created in Part 1 and the JSON file updated in Part 2. From these datasets, you will generate three visualizations to display the data: two dynamic graphs and one static graph.

Note: This Web Application is independent of the Web App you made for Lab01

Lab Rules

You may **not** use any kind of generators or website/app builders other than streamlit itself.

All code written in this assignment must be written by you. Since all students will be following the same tutorial for Part 1, we anticipate that your code will be similar; however, we do expect you to customize it. We are also expecting completely unique content for Part 2. While we encourage the

use of Google to reference streamlit properties, any **large sections** of code taken from another source are not permitted (including the projects listed on Streamlit's website).

Submissions violating any of these rules will receive an automatic 0 and a report to the Office of Student Integrity.

Getting Started

To start this lab, you will need the starter files provided on Canvas as `Lab02.zip`. Since streamlit is a Python library you can use IDLE or another editor of your choice to modify the template files and create your project.

Starter Files

To start this lab, download the `Lab02.zip` file from Canvas. Extract the files by right-clicking the `.zip` and clicking "Extract All" (for Windows) or double-clicking the file on Mac. Once you extract the files, you should see the following file structure:

```
Lab02
- HomePage.py # File for a brief introduction to your project
- requirements.txt # File for all modules that have to be installed
  - pages # Folder that stores your sidebar files
    - Survey.py # File for your data collection page that will write to data.csv
    - Visuals.py # File for your graphs page
- data.json # File that you will use to store your data for Part 2
- data.csv # File that stores information entered in Survey.py
```

Use these files as a base to start Part 1 of your lab. Make sure that all images you add from your computer are inside the **images** folder shown above. **All files related to your lab must be in this Lab02 folder.**

Streamlit

What is Streamlit?

Streamlit is specifically designed for data scientists, engineers, and developers to quickly turn data scripts into shareable web apps. Streamlit allows you to build interactive web applications using Python code, and you don't need extensive web development experience to get started.

Running Streamlit Scripts

To run your code on streamlit, you must execute your script directly from the command line using Command Prompt (Windows) or Terminal (Mac). Launch the Command Prompt/Terminal application on your computer.

Let's say your files are saved in your documents in the CS 1301 folder in your computer. First, you need to move command line into that folder (also called a directory) by typing the following line into your Command Prompt or Terminal:

- `cd Documents/CS 1301` (Or wherever your Lab02 files are located)

Run the script by typing the following line into the Command Prompt or Terminal and pressing `ENTER`, replacing `script_name.py` with the name of the file you want to execute, e.g. `HomePage.py`. Make sure your script name is in quotes.

- `streamlit run "script_name.py"`

To exit the `streamlit` application, click onto the Command Prompt or Terminal and type the following on your keyboard:

- `ctrl + c`

Additional Resources

Streamlit's official documentation is the most comprehensive and authoritative source for learning Streamlit. It provides in-depth guides, tutorials, and examples, making it the perfect starting point for beginners and advanced users. [Documentation](#)

Streamlit's official YouTube channel is a valuable resource for visual learners. It features video tutorials and demonstrations that help you understand how to create web apps with Streamlit. [Channel](#)

Additionally, the Lab Team have implemented an **EXAMPLE** of Lab02 to help inspire students. This is a starting place, and not something to copy. Additionally, do not use data about Sleep Averages per Week so it is not too similar to our example! Keep in mind that the example has more data visualisations than is required. They are there to help give you an idea of different types of data visualisation graphs. Check it out [here](#) Link: <https://lab02example-hsahutvqf3daqbiu7nywf2.streamlit.app/>

Part 1: Writing to a CSV File

In Part 1, you will create and write to a CSV file through your Streamlit app. You should use the file provided, `Survey.py`. This page should do the following:

- Accept input from the user in the form of a string
 - The input can be some form of a question. For example, "How much sleep do you get each night?". Then, the user can type in their responses which are then formatted to the CSV.
- Be able to accept enough data such that you can make a graph
- Write the inputted data to `data.csv`

Part 2: Making a JSON

In Part 2, you will be updating the provided JSON file. You are given a basic skeleton file for how to format your JSON, though it should be straightforward. Essentially, you will be creating a dictionary to read from. You may create this one of two ways:

- Manual Entry: Write your own data directly into a dictionary. This option may be simpler if you already have an idea of the data you want to include.
- Data Import: Find an existing dataset and export or copy its JSON structure into your file.

Part 3: Reading Data and Making Graphs

In Part 3, you will build a **new** page to display the data about yourself (or your fictional character) using the `.csv` and `.json` files you created in Parts 1 and 2. This page should be implemented in the file `Graphs.py`.

The goal for this part is to create a **functional page** that allows you to effectively visualise your data. Your page must meet the following requirements:

1. Data Visualizations

- You must include **3 graphs in total**
- At least **2** must be dynamic (explained below); **1** may be static
- Each graph must be a different type (e.g., bar chart, line chart, scatter plot)
- Each graph must include a short description that explains what the graph shows

2. User Interaction

- Your app must include at least **2** user inputs or interactions (e.g., text inputs, sliders, select boxes) that affect the app's behavior
- Inputs that serve no purpose (e.g., a button or slider with no effect) will **not** count

3. Session States

- You must make use of Streamlit [Session States](#) with your dynamic graphs. This is so data can be changed without refreshing the entire website. Check the Canvas Modules as well for more information. This should not be trivial. (In the example given, session states are used to allow the graphs to be changed repeatedly). These will help with dynamic graphs.

4. Dependencies

- If you use modules that are not part of the Python Standard Library (`math` , `random` , etc.), you **must** list out all of the modules that must be installed for your app to work in `requirements.txt` , one module per line. Streamlit and Pandas are already included in the `requirements.txt` file that is provided.

5. Data Sources

- At least one graph must use data from your `.csv` file and at least one graph must use data from your `.json` file
- The third graph may use either file type

App Deployment

Before we submit the lab, deploy your web app to streamlit's website so it can be viewed online. **You should have already done this for Lab01.** For help, refer back to the **Web Development Lab01 module on Canvas** for a tutorial on how to deploy your web app.

At the end of the lab, you will submit a link to your deployed web application *and* a zipped folder of your code to Gradescope.

Extra Credit - Early Submission

You may receive up to **10 bonus points** on this lab (allowing for a max grade of 110/100 on Lab02) for the following:

Requirement: Early Submission (10 points)

You will receive extra credit if you submit your completed lab before **Monday, October 13th, at 11:59 PM**. To receive extra credit, submit to the same Web Development Lab02 assignment on Gradescope before the early deadline. You will be disqualified from earning the extra credit if you re-submit after Monday, October 13th, at 11:59 PM.

Submission Process

Lab02 will be submitted through Gradescope. You will need to submit all files used in Lab02 (all images and python files) as well as a link to your deployed. To do this, compress the entire Lab02 folder that you are working in into a `.zip` file. Follow the instructions below:

1. Right click your Lab02 folder:
 - On Windows, click **Send To > Compressed (zip) folder**
 - On Mac, click **Compress "Lab02"**
2. This will create a zip folder on your computer. Rename the zip folder so that it has the following format (replace the below with your first and last name):

George-Burdell.zip

3. Upload this `.zip` file to the Lab02 assignment on **Gradescope**. After that, **you will also need to upload a link to your deployed website**, so make sure to submit a link to your website as well. To double check that you did this correctly, please download your submission and run your app to ensure everything works and looks correct.

Please put all images that you used in your website in this Lab02 folder. Otherwise, your grader will not be able to see images that you included.

Grading Rubric

Requirement	Points
Graph 1: Static	20
Graph 2: Dynamic	30
Graph 3: Dynamic	30
CSV Input	20
Extra Credit	10
Total	110/100

Final Total (with extra credit): 110/100 points